Office de la propriété intellectuelle du Canada

Canadian Intellectual Property Office

Un organisme d'Industrie Canada

An Agency of Industry Canada
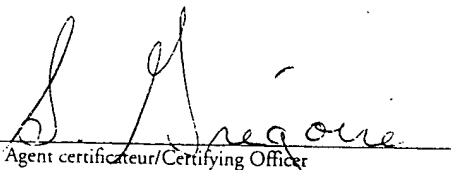
Bureau canadien des brevets

Certification

Canadian Patent Office

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,335,540**, on February 9, 2001, by **STERGIOS V. ANASTASIADIS**, for "Server Based Smoothing of Variable Bit Rate Streams".

CERTIFIED COPY OF PRIORITY DOCUMENT

Agent certificateur/Certifying Officer

February 18, 2002

Date

Canadä

(CIPO 68)
01-12-00

OPIC   CIPO

Bureau canadien des brevets

Certification

Canadian Patent Office

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,335,540**, on February 9, 2001, by **STERGIOS V. ANASTASIADIS**, for "Server Based Smoothing of Variable Bit Rate Streams".

CERTIFIED COPY OF PRIORITY DOCUMENT

_____
Agent certificateur/Certifying Officer

February 18, 2002
_____
Date

Canada

(CIPO 68)
01-12-00

OPIC CIPO

Server-Based Smoothing of Variable Bit Rate Streams

Field of Invention

5   This invention relates to continuous media servers and in particular to a server-based system and method for smoothing of variable bit-rate streams.

Background of the Invention

10   There is a growing demand for flexible and efficient access to collections of diverse video material from large-scale video servers. While prior designs frequently relied on significant resources and capabilities at the client, emergence of client devices with widely different hardware configurations make it necessary to reconsider such assumptions. Previously proposed techniques for smoothing variable bit rate streams

15   typically required extra buffer space at the client for prefetching stream data.

Variable bit rate encoding of video streams can achieve quality equivalent to constant bit rate encoding while requiring average bit rate that is lower by 40%. However, variable bit rate streams have high variability in their resource requirements which can

20   lead to low utilization of disk and network bandwidth in the common case. This occurs because the aggregate bandwidth requirements of concurrently served streams can be significantly higher than on average at particular time instances, and the admission control process bases its decisions on peak aggregate demand when considering new stream requests.

25

In order to improve resource utilization and the throughput of the system, a number of smoothing techniques have been proposed that can remove peaks in the required transfer bandwidth of individual streams by appropriately prefetching stream data at times of lower bandwidth demand. To date smoothing schemes always prefetched data into the

30   client buffers. Although such an approach can improve the utilization of both disk and network bandwidth, it is dependent on the amount of buffer space available at the client.

Several smoothing techniques deal with network link transfers of stored video streams. Salehi et al. describe a network smoothing technique to minimize the variability

of network bandwidth requirements assuming a fixed-size client buffer. Feng and Rexford compare the scheme of Salehi et al. with alternative schemes that minimize the total number of network bandwidth increases or decreases. McManus and Ross introduce a dynamic programming methodology for scheduling network transfers. Zhao and Tripathi

5   describe a class of methods that minimize the maximum required network bandwidth when multiplexing stream network transfers to multiple clients.

Other related research tries to improve network link utilization for live video. Rexford et al. use client data prefetching for smoothing live video streams, where the

10   stream requirements are known only for a limited period of time instead of the entire playback period. Mansour et al. examine several resource tradeoffs in live video smoothing. Ideas from live video smoothing are combined with prefix caching for smoothing streams in network proxies, assuming extra knowledge of quality of service parameters about the client not usually available at the server. This assumption about

15   limited knowledge of client resources at the server is a major disadvantage with client based smoothing methods.

Several studies have considered server-side resource management. Patterson et al. apply a cost-benefit analysis in order to control the disk bandwidth versus data buffer size

20   tradeoff for general applications. Paek and Chang propose an approach that, given a set of streams, optimizes a ``general objective function" by controlling the maximum disk bandwidth and buffer space available to each stream. Reddy and Wijayaratne have evaluated the effect of client-based smoothing on alternative disk striping methods; they point out the need for also studying server-based prefetching techniques in their future

25   work. Other schemes that have been proposed require that a certain amount of data be retrieved into the server buffer before playback can start, which reduces responsiveness. In addition, previous studies are limited to single disk systems with fixed size transfers only.

Kim et al. outline some ideas on how to control the tradeoff between buffer and

30   disk bandwidth utilization in stored video streaming. Their work differs from ours in several aspects, and they specify no concrete method for the problem. Their approach divides the stream into arbitrary length segments according to an ``empirical threshold", alpha. Prefetching is done only within a segment, and it is controlled by an ``empirical threshold", beta. Their disk bandwidth definition ignores the disk arm movement delays

and the round duration, and their simulation study is limited to single disk systems only. It is therefore an aspect of an object of the present invention for providing a smoothing method that prevents the required proportion of server buffer from exceeding the correspondingly decreased proportion of the required disk bandwidth.

Other have also addressed efficient retrieval of stream data from heterogeneous disks. Dan and Sitaram suggest that multiple heterogeneous storage devices may coexist in a video server environment. Considering the complexity of striping data across all the devices, they propose clustering homogeneous devices into groups and describe a dynamic data placement policy to keep the bandwidth and storage space utilization high. Chou et al. propose dynamic object replication techniques for improving utilization across different groups of disks. Santos and Muntz use randomized replication techniques for load balancing of heterogeneous disk arrays, while other studies try to achieve that with alternative disk array organizations and striping methods. Unlike all these methods, which are applicable (or have been demonstrated to work) only for constant rate streams, It is therefore an aspect of an object of the present invention a smoothing method for efficiently striping variable bit rate streams across heterogeneous disks.

Summary of the Invention

According to an aspect of the invention, there is provided a server-based smoothing method that uses only buffer space available at the server for smoothing disk data transfers. This method was incorporated into a prototype server, and demonstrate improved disk bandwidth utilization and considerable increase in the number of streams that can be supported at different system scales. The smoothing method has also been extended to support striping of variable bit rate streams across heterogeneous disks, and show an improvement in the server throughput that exceeds a factor of three at high loads.

The present invention is to maximize the average number of users supported concurrently in video server systems, by applying smoothing techniques and combining them appropriately with disk striping and admission control policies. Thus, the stream smoothing method that prefetches data into server buffers, which has several important advantages: ability to support clients with minimal memory resources (such as inexpensive mass-produced specialized devices) and still benefit from smoothing, optimizing for disk bandwidth, which is estimated to increase at rates an order of magnitude slower than

network link bandwidth, reduced complexity in admission control processing, by not having to support a separate transfer schedule for each individual client type, and reduced stream replication with disk striping policies that are based on a specified client configuration and retrieval sequence.

In order to prevent excessive smoothing from exhausting the available buffer space, the method applies a novel scheme where data prefetching is done as long as the proportion of server buffer required by each stream does not exceed the corresponding (decreased) proportion of the required disk bandwidth. Thus, the smoothing process is adjusted automatically, according to the total memory and disk bandwidth available in the server.

According to another aspect of the invention, there is provided smoothing of variable bit rate streams striped across heterogeneous disks. In the past, it was assumed that load-balancing and reliability problems restrict the size of disk arrays across which stream data can be striped efficiently. However, more recently disk striping schemes that are scalable have been introduced. It is important to consider the efficient operation of a server with heterogeneous disks because this allows server installations to be incrementally expanded using the most advanced and cost efficient storage devices as the system load increases. With the ratio between disk storage capacity and disk bandwidth increasing by a factor of ten every decade, disk accesses are becoming more precious. Therefore, bandwidth is the particular disk resource that the present invention strives to use best.

Brief Description of the Drawings

The accompanying drawings: Figures 1 to 13.

Detailed Description of the Preferred Embodiments

The present invention operates according to the server-push model as illustrated in Figure 1. When a playback session starts, the server periodically sends data to the client

until either the end of the stream is reached, or the client explicitly requests suspension of the playback. Data transfers occur in rounds of fixed duration Tround. In each round, an appropriate amount of data is retrieved from the disks into a set of server buffers reserved for each active client. Concurrently, data are sent from the server buffers to the client
5 through the network interfaces.

The amount of stream data periodically sent to the client is determined by the decoding frame rate of the stream and the resource management policy of the network. A policy is to set to send to the client during each round the amount of data that is needed for
10 the decoding process at the client in the next round; any other policy that does not violate the timing requirements and buffering constraints of the decoding client would be also acceptable.

The streams are compressed according to the MPEG-2 specification, or any other
15 encoding scheme that supports constant quality quantization parameters and variable bit rates. The stream data are stored across multiple disks, as shown in Figure 1. Playback requests arriving from the clients are initially directed to an admission control module, where it is determined whether enough resources exist to activate the requested playback session either immediately or within a limited number of rounds. A schedule database
20 maintains for each stream information on how much data needs to be accessed from each disk in any given round, the amount of server buffer space required, and how much data needs to be transferred to the client. This scheduling information is generated when the media stream is first stored and is used for both admission control and to control data transfers during playback. It is possible that two or more replicas are available for each
25 stream file, with different corresponding schedules.

In accordance with an embodiment of the present invention, there is provided a method called stride-based allocation for disk space allocation. In stride-based allocation, disk space is allocated in large, fixed-sized chunks called strides, which are chosen larger
30 than the maximum stream request size per disk during a round. This form of allocation is advantageous in the present invention, because stored streams are accessed sequentially according to a predefined (albeit variable) rate; therefore, the maximum amount of data accessed from a disk during a round for a stream is known a priori. When a stream is retrieved, only the requested amount of data is fetched to memory, and not the entire

stride. A stride may contain data of more than one round. One advantage of stride-based allocation is that it sets an upper-bound on the estimated disk access overhead during retrieval: since the size of a stream request never exceeds the stride size during a round, at most two partial stride accesses will be required to serve the request of a round on each

5   disk.

A mathematical abstraction of the resource requirements is necessary for scheduling purposes. Initially, consider a system with D functionally equivalent disks. A more general case of heterogeneous environments is examined later. In the following

10   sequence definitions, a zero value is assumed outside the specified range.

The stream Network Sequence, Sn, of length Ln defines the amount of data, Sn(i), 1=< i=< Ln, that the server must send to a particular client during round i of its playback. The Buffer Sequence, Sb, of length Lb = Ln + 1 defines the server buffer space, Sb(i), 0=< i=<

15   Lb, required by the stream data during round i. The Disk Sequence Sd of length Ld = Ln defines the total amount of data, Sd(i), 0=< i=< Ld - 1, retrieved from all the disks in round i for the client. Assuming that data are stored on the disks in logical blocks of fixed size Bl, which is multiple of the physical sector size Bp of the disk. Both the disk transfer requests and the memory buffer reservations are specified in multiples of the block size Bl.

20   The Disk Striping Sequence Sm of length Ld determines the amount of data Sm(i,k), 0=< i=< Ld - 1, that are retrieved from disk k, 0=< k =< D-1, in round i.

The disk sequence Sd can be derived from the network sequence Sn as follows: If

25

$$K^d(i) = \left\lceil \frac{\sum_{0 \le j \le i} S_n(j+1)}{B_l} \right\rceil$$

specifies the cumulative number of blocks Bl retrieved through round i, then

30

$$S_d(i) = (K^d(i) - K^d(i-1)) \cdot B_l.$$

Subsequently, the disk striping sequence Sm can be generated from the disk sequence Sd and the striping policy used.

Assuming that each disk has edge to edge seek time TfullSeek, single track seek time TtrackSeek, average rotational latency TavgRot, and minimum internal transfer rate Rdisk. The stride-based disk space allocation policy enforces an upper bound of at most two disk arm movements per disk for each client per round. The total seek distance can also be limited using a CSCAN disk scheduling policy. Let Mi be the number of active streams during round i of the system operation, and lj the round of system operation that the playback of stream j, $1 =< j =< M_i$, started. Then, the total access time on disk k in round i of the system operation has an upper-bound of:

$$T_{disk}(i, k) = 2T_{fullSeek} + 2M_i \cdot (T_{trackSeek} + T_{avgRot})$$
$$+ \sum_{j=1}^{M_i} S_m^j(i - l_j, k)/R_{disk}$$

where S_m^j is the disk striping sequence of client j. TfullSeek is counted twice due to the disk arm movement from the CSCAN policy, while the factor two in the second term is due to the stride-based allocation scheme we use. The first term should be accounted for only once in the disk time reservation structure of each disk, but each client j incurs an extra access time of

$$T_{disk}^j(i, k) = 2 \cdot (T_{trackSeek} + T_{avgRot})$$
$$+ S_m^j(i - l_j, k)/R_{disk}$$

on disk k during round i, when $S_m^j(i-l_j,k) > 0$, and zero otherwise.

If Rnet is the total network bandwidth available to the server, then the corresponding network transmission time reserved for client j in round i becomes $T^j net(i) = S_n^j(i-l_j) / R_{net}$, where Sn is the network sequence of client j. Also, the total server memory buffer reserved for client j in round i becomes $B^j(i) = S_b^j(i-l_j)$, where Sb is the buffer sequence of client j.

In Variable-Grain Striping, the disk striping sequence is defined as follows:

$$S_m^v(i,k) = (K^v(i) - K^v(i-1)) \cdot B_l,$$

when $i \bmod D = k$, with

$$K^v(i) = \left\lceil \frac{\sum_{0 \leq j \leq i} S_d(j)}{B_l} \right\rceil,$$

and $S_m{}^{\wedge}v(i,k)=0$ when $i \bmod D$ not equal k. Intuitively, the data retrieved during a round for a client are always accessed from a single disk, and the disks are used in round-robin fashion in successive rounds. The disk sequence thus determines the particular single disk accessed and the exact amount of data retrieved during each round. Comparison with alternative striping techniques has shown significant performance benefit for Variable-Grain Striping, and this is the method that is used.

In addition, since disk striping becomes more efficient with a priori knowledge of the retrieval process, it is useful to investigate alternative techniques that depend on a known server configuration only. Even without extra client buffers, efficient use of the disk bandwidth remains critical, as recent technological trends show that disk bandwidth increases much slower than network link bandwidth. Server-based prefetching deals with disk bandwidth and does not alleviate the network utilization problem. Thus, its operation can be complemented with network smoothing techniques for cases where sufficient client buffer can be assumed.

Previous studies have pointed out the potentially low disk utilization (and system throughput) achieved when retrieving variable-bit rate streams, and the need for appropriately prefetching data into the server buffers. A similar utilization problem was also studied in the context of network links carrying variable bit rate streams, where it was proposed to smooth bit rate peaks along a stream by prefetching data into client buffers. It was shown that such an approach can improve bandwidth utilization in both disks and network links, but is dependent on the memory configuration of individual clients.

Here, the smoothing of out disk bandwidth peaks is handled by prefetching stream data into server buffers. One crucial issue with disk prefetching is how to maintain an appropriate balance between disk bandwidth and server buffer space usage. Too aggressive prefetching can limit the number of concurrent streams that can be supported

5 because of excessive server buffer usage. Existing client-based smoothing methods do not have this problem, due to their implicit assumption of a fixed available client buffer size, and the fact that the client buffer space need not be multiplexed among different streams as is the case when buffering is done at the server.

10 In accordance with an embodiment of the present invention, there is provided a stream scheduling procedure that specifies for each stream both the variable server buffer and disk bandwidth requirements over time. A disk block b originally scheduled for round i is prefetched in a previous round j only if: i) the disk bandwidth requirement in round j with the prefetched block does not exceed the original disk bandwidth requirement of

15 round i, and ii) the proportion of server buffer required in each of the rounds j up to i-1 after prefetching block b does not exceed the proportion of disk bandwidth required in round i without b. The first condition is necessary in order for the prefetching to have a smoothing effect on the disk bandwidth requirements over time. The second condition is a heuristic that is applied in order to prevent exhaustion of the server buffer. Both

20 conditions are applied to individual streams, and to multiple streams concurrently.

A ``smoothness'' criterion that is based on Majorization Theory is used. For any $x=(x1, \ldots ,xn)$ in R^n, let the square bracket subscripts denote the elements of x in decreasing order $x[1] >= \ldots >= x[n]$. For x,y in R^n, x is majorized by y, $x < y$, if:

25

$$\sum_{i=1}^{k} x_{[i]} \le \sum_{i=1}^{k} y_{[i]}, \quad k = 1, \cdots, n-1$$

30   and

$$\sum_{i=1}^{n} x_{[i]} = \sum_{i=1}^{n} y_{[i]}.$$

Then, consider x smoother than y, if x < y. Finally, we call a vector x in R^n majorization-minimal if there is no other vector z in R^n such that z < x.

In accordance with an embodiment of the present invention, there is provided a method that, given a stream network sequence Sn and a target server configuration, generates a smoothed disk sequence Sd. The generated disk sequence is majorization-minimal under the specified constraints. The generated disk sequence is subsequently transformed into a striping sequence Sm according to some disk striping method, such as the Variable-Grain Striping.

The disk time reservation for a disk transfer of X bytes is approximately equal to:

$$T_{disk}(X) = 2 \cdot (T_{trackSeek} + T_{avgRot}) + X/R_{disk}.$$

Definition 1: Let the Disk Time Proportion of X bytes, Pd(X), be the fraction of the round time Tround that the disk time reservation Tdisk(X) occupies: Pd(X)=Tdisk(X)/Tround. Further let the Buffer Space Proportion of X bytes, Pb(X), be the fraction of the buffer space for each disk, Bdisk, (Bdisk is the total server buffer divided by the number of disks D) that X bytes occupy in a round: Pb(X)=X/Bdisk. Then, the Maximum Resource Proportion in round i, is the maximum of the corresponding disk time and buffer space proportions in that round: {\small max(Pd(Sd(i)),Pb(Sb(i))).

The above utilization definitions refer to resource reservations within a round. However, for conciseness of the following presentation, the words Round and Reserved are dropped and are refer to as Disk Utilization, Buffer Utilization and Maximum Utilization, respectively.

Definition 2: The Deadline Round for a block is the latest round at which the block can be accessed from the disk without incurring a real-time violation at the network transfer. Then, with respect to a specific block, all rounds before the deadline round are considered Candidate Rounds and the one actually chosen for prefetching is called the Prefetch Round. All the rounds between the deadline and a prefetch round are called Shift Rounds.

These definitions only affect the number of blocks accessed in each round, since stream block accesses are done sequentially during playback.

Definition 3: The Maximum-Proportion Constraint is defined as the requirement that the maximum resource proportion of the deadline round is no less than the maximum resource proportion of the corresponding (if any) prefetch and shift rounds.

The Server Smoothing method is as follows:

```
0.  proc serverSmoothing
1.  input : L_n, S_n[] ( =0 outside [1..L_d] ), B_l
2.  output : L_d, S_d[], L_b, S_b[]
3.  begin
4.    blockQuantize(L_n, S_n[], B_l)  (* see App. B *)
5.    for t_rnd : 0..L_n-1
6.      if ( P_buf(S_b(t_rnd)) < P_dsk(S_d(t_rnd)) )
7.        repeat
8.          t_min := t_rnd
9.          P_min := max( P_dsk(S_d(t_rnd)), P_buf(S_b(t_rnd)) )
10.         t_prv := t_rnd - 1, prefFailed := false
11.         while ( prefFailed = false AND t_prv >= 0 )
12.           P_prf = max( P_dsk(S_d(t_prv) + B_l),
13.                        P_buf(S_b(t_prv) + B_l) )
14.           P_shf = max( P_dsk(S_d(t_prv)),
15.                        P_buf(S_b(t_prv) + B_l) )
16.           (*check for max proportion decrease*)
17.           if (P_prf < P_min)
18.             t_min = t_prv, P_min = P_prf
19.           else if (P_min < P_shf)
20.             prefFailed := true
21.           end-if
22.           t_prv := t_prv - 1
23.         end-while
24.         if (t_min < t_rnd)  (* update vectors *)
25.           S_d(t_min) := S_d(t_min) + B_l,
26.           S_b(t_min) := S_b(t_min) + B_l
27.           for t_prv := t_min + 1 .. t_rnd - 1
28.             S_b(t_prv) := S_b(t_prv) + B_l
29.           end-for
30.           S_d(t_rnd) := S_d(t_rnd) - B_l
31.         end-if
32.       until (t_min >= t_rnd) (*prefetch search failed*)
33.      end-if
34.    end-for
35.  end
```

This method initially invokes the blockQuantize procedure (see appendix B) that generates disk and buffer sequences with data transfer sizes that are integral multiples of the logical block Bl. Network transfers are specified in byte granularity for increased flexibility (if necessary, they could be quantized too). Then, rounds of the generated sequences are visited in increasing order starting from round zero. For every logical block to be retrieved from the disk in the currently visited round, previous rounds are examined linearly in decreasing order towards round zero for potential prefetching of the block. Each such search completes successfully when a prefetch round is found such that the maximum resource proportion of the current round decreases while remaining higher than those of the prefetch and shift rounds.

12

This implies that the disk sequence can be smoothed out without incurring buffer space proportion peaks exceeding the disk time proportion of the current round. Otherwise, the block prefetching operation will not have any positive smoothing effect

.5 overall, and the corresponding search fails.

The method is shown to work correctly below:

Lemma 1: The Server Smoothing method chooses the prefetch round for each block in a

10 way that satisfies the following properties: i) No network transfer timing violation occurs. ii) No maximum-proportion constraint violation occurs. iii) It has the lowest possible disk time proportion. iv) It is closest to the deadline round. Property (iii) prevails when in conflict with property (iv). Proof: Available in appendix A.

Definition 4 If $\beta_1 \geq \cdots \geq \beta_n$ are integers and $\beta_i >$ β_j, then the transformation $\beta_i' = \beta_i - 1$, $\beta_j' = \beta_j + 1$, $\beta_k' = \beta_k$, for all $k \neq i, j$ is called a *transfer from i to* j.[5]

Lemma 2 *(Muirhead, 1903 [18]) If* $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots,$ β_n *are integers and* $\alpha \prec \beta$, *then* α *can be derived from*

β *by successive applications of a finite number of transfers.*

Proof: See Marshall and Olkin [16], pg. 135.□

In the presentation that follows *transfer* units correspond to logical blocks of size B_l as opposed to individual bytes.

Lemma 3 *The Server Smoothing algorithm produces disk sequence that satisfies the properties of Lemma 1, and has no transfer that would not violate them.*

Proof: Available in appendix A.

Theorem: 1 *The Server Smoothing algorithm generates a majorization-minimal disk sequence that satisfies the properties of Lemma 1.*

Proof: From Lemma 3, the disk sequence generated by the Server Smoothing algorithm satisfies the properties of Lemma 1 and has no *transfer* that would not violate them. Then, from Lemma 2, there is no other sequence that satisfies the properties of Lemma 1 and is majorized by the disk sequence generated by the Server Smoothing algorithm. If such a sequence existed, additional block *transfers* would be acceptable.□

The computational complexity of the Server Smoothing algorithm is $O(\frac{\sum_{i=1}^{L_n} S_n(i)}{B_l} L_n^2)$, where S_n is the input

network sequence and Ln is its length. Although it may be possible to reduce this complexity, practically the application of this method on a video stream of 30 minutes completes in tens of seconds on a 133MHz RISC processor with Bl=16 KB, Ln=1,800 and

$$\sum_{i=1}^{L_n} S_n(i) = 1.12 \cdot 10^9.$$

Since the schedule generation is done off-line, the above execution time is acceptable. The higher computational complexity relative to the O(Ln) complexity of network smoothing methods is the extra cost for avoiding the ``hardwired" fixed client buffer constraint. The Server Smoothing method generates majorization-minimal disk sequence with several buffer constraints, including the fixed buffer of network smoothing methods as a special case.

A more complete understanding of the invention can be obtained by reference to the following specific Implementations. These Implementations are described solely for purposes of illustration and are not intended to limit the scope of the invention. Changes in form and substitution of equivalents are contemplated as circumstances may suggest or render expedient. Although specific terms have been employed herein, such terms are intended in a descriptive sense and not for purposes of limitation.

Implementations

The Implementations are described for the purposes of illustration and are not intended to limit the scope of the invention.

A media server evaluationation system has been designed and built, in order to evaluate the resource requirements of alternative stream scheduling techniques. The modules are implemented in about 12,000 lines of C++/ Pthreads code on AIX4.1. The code is linked either to the University of Michigan DiskSim disk simulation package, which incorporates advanced features of modern disks such as on-disk cache and zones for simulated disk access time measurements, or to hardware disks through their raw device interfaces. The indexing metadata are stored as regular Unix files, and during operation are

kept in main memory. A MPEG-2 decoder is used from the MPEG Software Simulation Group for stream frame size identification.

The basic responsibilities of the media server include file naming, resource

5   reservation, admission control, logical to physical metadata mapping, buffer management, and disk and network transfer scheduling.

With appropriate configuration parameters, the system can operate at different levels of detail. In Admission Control mode, the system receives playback requests, does

10   admission control and resource reservation, but no actual data transfers take place. In Simulated Disk mode, all the modules become functional, and disk request processing takes place using the specified DiskSim disk array. There is also the Full Operation mode, where the system accesses hardware disks and transfers data to fixed client network addresses.

15

Assuming that playback initiation requests arrive independently of one another, according to a Poisson process. The system load can be controlled through the arrival rate λ of playback initiation requests. Assuming that the disk transfers form the bottleneck resource and consider the ideal system, with no disk overhead when accessing disk data,

20   as "perfectly efficient system", then, setting the maximum arrival rate $\lambda = \lambda max$ of playback requests equal to the mean stream completion rate in that perfectly efficient system. This creates enough system load to show the performance benefit of arbitrarily efficient data striping policies. The mean stream completion rate μ, expressed in streams per round, for streams of average data size Stot bytes becomes:

25

$$\mu = \frac{D \ disks \cdot R_{disk} \ \frac{MB/s}{disk} \cdot T_{round} \ \frac{s}{round}}{S_{tot} \ \frac{MB}{stream}}$$

$$= \frac{D \cdot R_{disk} \cdot T_{round}}{S_{tot}} \ \frac{streams}{round}.$$

30

The corresponding system load becomes: $\rho = \frac{\lambda}{\mu} \leq 1$, where $\lambda \leq \lambda_{max} = \mu$.

In the admission control process, when a playback request arrives, it is checked whether available resources exist for every round during playback. The test considers the exact data transfers of the requested playback for every round and also the corresponding available disk transfer time, network transfer time and buffer space in the system. If the request cannot be initiated in the next round, the test is repeated for each round up to [1/λ] rounds into the future, until the first round is found where the requested playback can be started with guaranteed sufficiency of resources. Checking [1/λ] rounds into the future achieves most of the potential system capacity as was shown previously. If not accepted, the request is rejected rather than being kept in a queue.

As the basic performance metric (basic benchmark), the average number of active playback sessions that can be supported by the server was chosen. The objective is to make this number as high as possible.

Six different VBR MPEG-2 streams of 30 minutes duration each were used. Each stream has 54,000 frames with a resolution of 720x480 and 24 bit color depth, 30 frames per second frequency, and a IBBPBBPBBPBBPBB 15 frame Group of Pictures structure. The encoding hardware used allowed the generated bit rate to take values between 1Mbit/s and 9.6Mbit/s. The statistical characteristics of the clips are given in Table 1. The coefficients of variation of bytes per round lie between 0.028 and 0.383, depending on the content type. In the basic benchmark, the six different streams are submitted round-robin. Where appropriate, results from individual stream types are also shown.

Table 1: Six MPEG-2 video streams of 30 minutes duration each. The coefficient of variation shown in the last column changes according to the content type.

Content Type	Avg Bytes per Round	Max Bytes per Round	CoV per Round
Science Fiction	624,935	1,201,221	0.383
Music Clip	624,728	1,201,221	0.366
Action	624,194	1,201,221	0.245
Talk Show	624,729	1,201,221	0.234
Adventure	624,658	1,201,221	0.201
Documentary	625,062	625,786	0.028

Although the Main Profile Main Level MPEG-2 specification allows bitrates up to 15Mbit/sec, there is a typical point of diminishing returns (no visual difference between original and compressed video) at 9Mbit/sec. The DVD specification sets a maximum allowed MPEG-2 bitrate of 9.8Mbit/sec.

5

For evaluation with homogeneous disks, Seagate Cheetah SCSI disks, with the features shown in Table 2 were assumed. Note that one megabyte (megabit) is considered equal to 2^{20} bytes (bits), except for the measurement of transmission rates and disk storage capacities where it is assumed equal to 10^6 bytes (bits) instead. Except for the storage capacity, which can reach 73GB in the latest models, the rest of the performance numbers are typical of today's high-end drives. The logical block size Bl was set to 16 KB bytes, while the physical sector size Bp was equal to 512 bytes. The stride size Bs in the disk space allocation was set to 2 MB. The server memory is organized in buffers of fixed size Bl=16 KB bytes each, with a typical total space of 256 MB for every extra disk. (The effect of buffer space is examined later.) The available network bandwidth was assumed to be infinite, leaving contention for the network outside the scope of the current work.

Table 2: Assumed features of the Seagate SCSI disk

Seagate Cheetah ST-34501N	
Data Bytes per Drive	4.55 GB
Average Sectors per Track	170
Data Cylinders	6,526
Data Surfaces	8
Zones	7
Buffer Size	0.5 MB
Track to Track Seek(read/write)	0.98/1.24 msec
Maximum Seek(read/write)	18.2/19.2 msec
Average Rotational Latency	2.99 msec
Internal Transfer Rate Inner Zone to Outer Zone Burst Inner Zone to Outer Zone Sustained	122 to 177 Mbit/s 11.3 to 16.8 MB/s

In the evaluations, the round time was set equal to one second. This round length was found to achieve most of the system capacity with reasonable initiation latency. This choice also facilitates comparison with previous work in which one second rounds were used. A warmup period of 3,000 rounds was used and calculated the average number of active streams from round 3,000 to round 9,000. The measurements were repeated until the half-length of the 95% confidence interval was within 5% of the estimated mean value of the number of active streams.

Starting with disk arrays consisting of functionally equivalent disks. In Figure 3, there is depicted the disk time and buffer space proportions in each round for a particular stream. Without smoothing (Fig. 3(a)), the occupied buffer space is the minimum necessary for data staging during disk and network transfers. With Server Smoothing (Fig. 3(b)), data are prefetched into the server buffer according to the maximum-proportion constraint. This keeps the maximum buffer space proportion to be no more than the maximum disk time proportion (8.7% in this example). The maximum disk time proportion drops from 11.5% to 8.7%, while the maximum buffer space proportion increases from less than 1% to 8.7%. The Seagate Cheetah disk parameters are assumed and server buffer of 256 MB per disk.

Referring to Figure 4, there is illustrated the number of active streams that is sustained at different system loads and array configurations with between 4 and 64 disks using the mixed workload. In all the cases shown, the stream data have been striped according to the Variable-Grain Striping method. The Server-Smoothed plots show the

performance benefit of applying the Server Smoothing method assuming 256 MB of available server buffer space for each extra disk. At moderate load ρ=50%, Variable-Grain Striping with no smoothing allows all stream requests to be accepted. At a higher load ρ=90% the Server Smoothing can improve throughput by over 10%. The corresponding

5 rejection rate (not shown) is 25% with Server Smoothing, and 41% with plain Variable Grain Striping, at 90% load. With the mixed workload and Variable-Grain Striping (with/without Server Smoothing), the sustained number of active streams increases almost linearly with the number of disks. Although at system load 50% all the submitted streams are accepted, at load 90% Server Smoothing increases the number of active streams by

10 about 10%. This benefit is maintained across different numbers of disks.

Results for individual stream types are shown in Figure 5. It was found that the benefit of Server Smoothing depends on the variability of data transfers across different rounds. Thus, although smoothing adds no benefit at streams with negligible variability

15 (e.g. Documentary), as variation becomes higher, the increase in the number of streams can exceed 15% (Action). The load was set to 90% on sixteen disks and 256 MB per disk were assumed.

In Figure 6, there is illustrated the average total reserved disk time Tdisk(i,k)

20 (expressed as percentage of round time) on a particular disk (k=0) during the measurement period 3,000 =< i < 9,000. While for most stream types the average disk time hardly exceeds 80% of the round time with plain Variable-Grain Striping, it consistently approaches 90% and in several cases exceeds 93% (Action, Music Clip, Talk Show) when Server Smoothing is applied. This is remarkably high when compared to the 96% achieved

25 by Documentary that has very low variation of data transfer sizes across different rounds. With sixteen disks and 90% system load, the average disk time reserved each round increases from less than 80% to over 90% with Server Smoothing and server buffer 256 MB per disk. Although the disk time shown corresponds to one of the disks (Disk 0) it was similar (typically within 2%) across the disks of the array.

30

In the evaluations until now, a server buffer of 256 MB per disk was assumed. Statistics gathered across the different stream workloads showed that the actual proportion of total occupied buffer space was about 50% on average with the maximum hardly

exceeding 60% at 90% load. Further increasing the aggregate buffer utilization, without the buffer becoming a potential bottleneck in the system, would require incorporating into the method information about the way stream requests are multiplexed.

5   From Figure 7, it is concluded that more than half of the Server Smoothing benefit is achieved with server buffer size as low as 64 MB per disk. The assumption of 256 MB server memory (per disk) in the smoothing process is still reasonable, however. The additional performance benefit from extra memory is sustained across different system sizes as was shown in Figure 4, with the purchase and administration cost of server
10   memory being only a fraction of the costs associated with high-end disk drives. With buffer space (per disk) set to 64 MB, more than half of the total benefit of Server Smoothing can be achieved (see also Figure 5). Increasing the buffer space to 256 MB farther improves the number of streams in Science Fiction and Action types although at a diminishing degree

15

It has traditionally been assumed that disk arrays consist of homogeneous disks, presumably in order to keep the system complexity manageable. With the scalability of stream striping demonstrated recently and the sequential access of stored video making things somewhat simpler, systems with different disk types were investigated that might
20   have been scaled incrementally with the newest disk technology. Newer disk models typically achieve higher transfer rates and have larger storage capacities.

The case of striping stream data across heterogeneous disk arrays was evaluated. The objective was to maximize the number of active streams by increasing the disk
25   bandwidth utilization across all the disks. This might lead to suboptimal storage capacity utilization, which is affordable given the current technology trends.

In the evaluations, it was assumed that disk arrays consists of Seagate (Table 2) and older HP disks in alternating order. The features of the HP disks are shown in Table 3.
30   A striking difference was noted in the minimum internal transfer rate, which is 2.8 MB/s for the HP disks, one fourth as much as the 11.3 MB/s of the Seagate disks. Such a difference only makes the balancing of the system load more challenging. Although the evaluations in this section assume an equal number of disks belonging to each type, other ratios in the number of disk types obtained similar results.

Table 3 Features of the HP SCSI disk that were included in the evaluations for heterogeneous environments

HP-C3323A	
Data Bytes per Drive	1,052,491,776
Data Sectors per Track	72 to 120
Data Cylinders	2,910
Data Surfaces	7
Zones	8
Buffer Size	0.5 MB
Track to Track Seek	< 2.5 msec
Maximum Seek	22 msec
Rotational Latency	5.56 msec +/- 0.5%
Internal Transfer Rate Inner to Outer Zone Burst Inner to Outer Zone Sustained	4.0 to 6.6 MB/s 2.8 to 4.7 MB/s

In Figure 8(a), there is depict an example of a stream striped across an heterogeneous disk array. The lower transfer rate of the HP disks creates peaks of disk time proportion that can exceed 40%. In order to alleviate this problem, the Server Smoothing method was extended to handle heterogeneous disks. In particular, the disk time Tdisk(X) and the disk time proportion function Pd(X) was redefined to accept a second disk type argument k that specifies the particular disk parameters to be used Pd(X, k)=Tdisk(X,k)/Tround. During the operation of the Server Smoothing method, the disk type k assumed in each round i can be derived using a simple rule, such as k=i(mod\ D), where D is the total number of the disks. Finally, if $R^{\wedge}kdisk$ is the minimum internal transfer rate of disk k, the service rate definition Equation becomes:

$$\mu = (T_{round} \cdot$$

$$\sum_{k=0}^{D-1} R^k_{disk})/S_{tot} \frac{streams}{round}.$$

When Server Smoothing is applied, disk transfers are appropriately adjusted to smooth out the peaks and keep the maximum reservation below 13%. At the same time, the maximum server buffer proportion is constrained to never exceed the maximum disk time proportion.

The extended Server Smoothing method was applied on the stream example of Figure 8(a). The generated transfer sequence, shown in Figure 8(b), has its buffer space

proportion bounded by the disk time proportion, as before. In addition the maximum disk time proportion dropped from over 40% to less than 13%, after the transfer sizes across different rounds were appropriately adjusted according to the available disk bandwidth.

5   In Figure 9, the performance of plain Variable-Grain Striping is compared to that of Variable-Grain Striping with Server Smoothing in a range of heterogeneous disks between 4 and 64. Although the number of streams always increases almost linearly with the number of disks, Server Smoothing achieved an advantage that exceeds a factor of 2 and 3 at loads of 50% and 90%, respectively. The reason is that the limited disk

10   bandwidth of the HP disks, prevents the Seagate disks from attaining high bandwidth utilization without appropriate adjustment of the disk transfers. A similar behavior is also demonstrated across different stream types in Figure 10. With plain Variable-Grain Striping, the number of supported streams on sixteen disks hardly exceeds 50; when Server Smoothing is added the number of streams gets close to 140. Server buffer space of

15   256 MB per disk has been assumed.

        In Figure 11, there is depicted the average time that the Seagate and HP disks are expected to be busy respectively during each round. The statistics are shown for disks 0 and 1 only, since the statistics for the rest of the disks were similar. The average time that

20   the Seagate disks are busy is less than 25% of the round time with plain Variable-Grain Striping. The reason is the low bandwidth of the HP disks, whose corresponding average time varies between 50% and 80%; it cannot get higher due to the relatively large data requirements of the individual streams. When Server Smoothing is applied, a high average reserved disk time that gets close to 90% is achieved across all the disks of the disk array.

25   This becomes possible with appropriate data prefetching that distributes data accesses across the disks according to the bandwidth that they can support.

        Referring to Figure 11, the two leftmost bars of each stream type, show the average reserved disk time for the Seagate (STN) and HP (HPC) disks, assuming plain Variable

30   Grain Striping and 90% load. The lower transfer bandwidth of the HP disk creates a bottleneck keeping the reserved disk time of the Seagate disk to less than 25% of the round time. As is shown by the two rightmost bars though, with Server Smoothing both disk types attain average disk time close to 90% of the round time.

In the previous evaluations, the server buffer were set to 256 MB per disk. However, seen from Figure 12, having only 64 MB per disk is sufficient to get most of the benefits of Server Smoothing for the particular streams included in our benchmark.

5   Referring to Figure 12, with the server buffer (per disk) set to 64 MB, most of the benefit of Server Smoothing can be attained (see also Figure 10). Increasing the server buffer from 64 MB to 256 MB increases only marginally (less than 5%) the sustained number of active streams.

10   Referring to Figure 13, in an array of four disks, Seagate (STN) and HP (HPC) models are used in alternating order. The average and maximum reserved and measured time is shown for the disks 0 and 1 of the array with the mixed workload at 90% load. On the STN disks, the reserved statistics are no more than 8% higher than the measured. On the HPC disk, the corresponding difference can get up to 20%. The measurements have
15   been done using the detailed disk simulation models by Ganger et al.

In order to keep the computation time reasonable, the previous evaluations were conducted with the system in Admission Control mode, where resource reservations are made for arriving playback requests, but without actual time measurement of the
20   individual disk transfers. In the present section, the Simulated Disk Mode was used to compare the statistics of the disk time reservations with the statistics gathered over the access times of all individual data transfers involved, using the DiskSim representation of the Seagate Cheetah and HP C3323A disks. A four-disk array model is used with the disk types in alternating order. Each disk is presumed to be attached to a separate 20 MB/sec
25   SCSI bus, and no contention is assumed on the host system bus connecting the SCSI buses. The statistics are gathered during 6,000 rounds after a warmup period of 3,000 rounds, as before and the mixed workload is used. The server can support 9.74 active streams with plain Variable-Grain Striping and 33.87 active streams with Server-Smoothed Variable-Grain Striping corresponding to a 90% load.
30

As can be seen from Figure 13, in both the average and maximum case, the reserved disk time is no more than 8% higher than the corresponding measurements on the Seagate disk model by Ganger et al. This gap can be attributed to the fact that the disk time reservation assumes a minimum disk transfer rate and ignores on-disk caching. The

corresponding gap for the HP disks gets close to 15% with plain striping and 20% with Server Smoothing. Possible reasons for the larger discrepancy with the HP disks are the increased on-disk cache locality due to the smaller disk capacity, and the higher probability that only one head movement is required with the smaller data transfers

5   (smoothed case).

In general, it is believed that the achieved accuracy in the disk time predicted by the resource reservation is adequate. In fact, to improve these estimates, it would probably be necessary to have extra disk geometry information that is not readily available for

10   commercial disk drives.

The recent interest in thin client devices, soon expected to outnumber powerful desktop computers, motivates the development of resource management policies that make minimal assumptions about available client resources. The Server Smoothing method for

15   variable bit rate streams that prefetch stream data into server buffers is advantageous for these developments.

Evaluation of homogeneous disk arrays and moderate server buffer space shows that Server Smoothing can achieve more than 15% increase in the number of streams that

20   can be supported by the server. This benefit is sustained across different disk numbers up to 64 disks that were examined.

Using the Server Smoothing method for striping variable bit rate streams across arrays of heterogeneous disks, it was demonstrated that, when plain disk striping is used,

25   disks with lower transfer rates prevent the system from reaching high utilization. When Server Smoothing is applied, the average reserved disk access time can get as high as 90% of the round time across the different disks. The corresponding benefit in the number of streams accepted by the server exceeded a factor of three for the particular disk array configuration that used.

30

One important issue that remains open is designing appropriate replication techniques for tolerating disk failures in heterogeneous disk environments.

Although preferred embodiments of the invention have been described herein, it will be understood by those skilled in the art that variations may be made thereto without departing from the scope of the invention.

Appendix A

Proof of Lemma 1: The property (i) comes from lines 10-11,22 of the method, which
5   limit the range of prefetching rounds to those preceding the current one. The property (ii)
is due to lines 17,19 which ensure that the maximum resource proportion of the deadline
round is no less than that of the prefetch and shift rounds respectively. The candidate
round with minimal disk time proportion (iii) is kept track of through variable Pmin in line
18. Finally, the closeness to the deadline (iv) is controlled by the descending loop at lines
10   22, and the strict inequality in line 17. \square

Proof of Lemma 3: The method is ``greedy" and using induction on the network sequence
length Ln. The generated disk sequence trivially satisfies the lemma claim at Ln=1, with
round 0 to access the data from disk and round 1 to send the data over the network.
15   Assuming that at Ln=k the claim is valid. It is shown that it is also valid for Ln=k+1. Let
us assume that we get the sequence of the k first disk accesses 0 ... k-1 to satisfy the
lemma claim, before starting to deal with the disk access of round k. Due to the property
(i) of Lemma 1, it is not possible to schedule in round k, block accesses from the previous
rounds. Therefore, the only acceptable transfer of blocks that remains is moving blocks of
20   the round k to previous rounds. An exhaustive search is done in the while-loop of the lines
11-23 of the method. Each of the previous rounds is visited, and a record is kept of the
closest round that can prefetch a block with minimal total disk time proportion from
property (iii). Property (ii) guarantees no violation of the maximum-proportion constraint.
The above search is repeated by the repeat-loop of lines 7-32 until no more transfers of
25   logical blocks belonging to round k are possible. (The decision to choose prefetch rounds
closest to the deadline round, from property (iv), leads to buffer occupancy minimization.)

<u>Appendix B</u>

B The blockQuantize procedure

```
0. proc blockQuantize
1. input : L_n, S_n[] ( =0 outside [1..L_n] ), B_l
2. output : L_d, S_d[], L_b, S_b[]
3. begin
4.    S_d[] := 0, S_b[] := 0
5.    L_d := L_n, L_b := L_n + 1
6.    totSn := 0
7.    for t_rad : 0..L_n
8.       pruSn := totSn, totSn := totSn + S_n(t_rad + 1)
9.       (* we use function ceil() for the ⌈ ⌉ operation *)
10.      S_d(t_rad) := B_l · ( ceil(totSn/B_l) − ceil(pruSn/B_l) )
11.      S_b(t_rad) := S_b(t_rad − 1) + S_d(t_rad) − S_n(t_rad − 1)
12.   end-for
13. end
```

28

References

[1] *MPEG-2 Encoder/Decoder, Version 1.2.* 1996. MPEG
Software Simulation Group.

[2] Anastasiadis, S. V., Sevcik, K. C., and Stumm, M.
Disk Striping Scalability in the Exedra Media Server.

In *ACM/SPIE Multimedia Computing and Networking Conf.* (San Jose, CA, Jan. 2001). (to appear).

[3] Biersack, E. W., and Hamdi, M. Cost-optimal Data Retrieval for Video Servers with Variable Bit Rate Video Streams. In *Intl. Work. Network and Operating System Support for Digital Audio and Video* (Cambridge, UK, July 1998), pp. 295–302.

[4] Bolosky, W. J., Barrera, J. S., Draves, R. P., Fitzgerald, R. P., Gibson, G. A., Jones, M. B., Levi, S. P., Myhrvold, N. P., and Rashid, R. F. The Tiger Video Fileserver. In *Intl. Work. on Network and Operating System Support for Digital Audio and Video* (Zushi, Japan, Apr. 1996), pp. 97–104.

[5] Chou, C. F., Golubchik, L., and Lui, J. C. S. A Performance Study of Dynamic Replication Techniques in Continuous Media Servers. In *ACM SIGMETRICS* (Atlanta, GA, May 1999), pp. 202–203.

[6] Dan, A., and Sitaram, D. An Online Video Placement Policy based on Bandwidth to Space Ratio (BSR). In *ACM SIGMOD* (San Jose, CA, May 1995), pp. 376–385.

[7] Feng, W.-C., and Rexford, J. A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video. In *IEEE INFOCOM* (Kobe, Japan, Apr. 1997), pp. 58–66.

[8] Ganger, G. R., Worthington, B. L., and Patt, Y. N. The DiskSim Simulation Environment: Version 2.0 Reference Manual. Tech. Rep. CSE-TR-358-98, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan, Dec. 1999.

[9] Gray, J., and Shenoy, P. Rules of Thumb in Data Engineering. In *IEEE Intl. Conf. Data Engineering* (San Diego, CA, Feb. 2000), pp. 3–10.

[10] Gringeri, S., Shuaib, K., Egorov, R., Lewis, A., Khasnabish, B., and Basch, B. Traffic Shaping, Bandwidth Allocation, and Quality Assessment for MPEG Video Distribution over Broadband Networks. IEEE Network, 6 (Nov/Dec 1998), 94–107.

[11] The IBM Dictionary of Computing. McGraw-Hill, New York, NY, 1994.

[12] Kim, I.-H., Kim, J.-W., Lee, S.-W., and Chung, K.-D. VBR Video Data Scheduling using Window-Based Prefetching. In IEEE Multimedia Computing and Systems (Florence, Italy, June 1999), pp. 159–164.

[13] Lee, D.-Y., and Yeom, H. Y. Tip Prefetching : Dealing with the bit rate variability of video streams. In IEEE Multimedia Computing and Systems (Florence, Italy, June 1999), pp. 352–356.

[14] Makaroff, D., Hutchinson, N., and Neufeld, G. An Evaluation of VBR Disk Admission Algorithms for Continuous Media File Servers. In ACM Multimedia (Seattle, WA, May 1997), pp. 143–154.

[15] Mansour, Y., Patt-Shamir, B., and Lapid, O. Optimal Smoothing Schedules for Real-Time Streams. In ACM Principles of Distributed Computing (Portland, OR, July 2000).

[16] Marshall, A. W., and Olkin, I. Inequalities: Theory of Majorization and its Applications. Academic Press, New York, 1979.

[17] McManus, J., and Ross, K. A Dynamic Programming Methodology for Managing Prerecorded VBR Sources in Packet-Switched Networks. Telecommunications Systems 9 (1998), 223–247.

[18] Muirhead, R. F. Some methods applicable to identities and inequalities of symmetric algebraic functions of n letters. In Proc. Edinburgh Mathematical Society (1903), vol. 21, pp. 144–157.

[19] Paek, S., and Chang, S.-F. Video Server Retrieval Scheduling for Variable Bit Rate Scalable Video. In IEEE Multimedia Computing and Systems (Hiroshima, Japan, June 1996), pp. 108–112.

[20] Patterson, R. H., Gibson, G. A., Ginting, E., Stodolsky, D., and Zelenka, J. Informed Prefetching and Caching. In ACM Symp. Operating Systems Principles (Copper Mountain Resort, CO, Dec. 1995), pp. 79–95.

[21] Reddy, A. L. N., and Wijayaratne, R. Techniques for improving the throughput of VBR streams. In ACM/SPIE Multimedia Computing and Networking (San Jose, CA, Jan. 1999), pp. 216–227.

[22] Rexford, J., Sen, S., Dey, J., Feng, W., Kurose, J., Stankovic, J., and Towsley, D. Online Smoothing of Live, Variable-Bit-Rate Video. IEEE Trans. on Multimedia 2, 1 (Mar. 2000), 37–48.

[23] Salehi, J. D., Zhang, Z.-L., Kurose, J. F., and Towsley, D. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. In ACM SIGMETRICS (Philadelphia, PA, May 1996), pp. 222–231.

[24] Santos, J. R., and Muntz, R. Performance Analysis of the RIO Multimedia Storage System with Heterogeneous Disk Configurations. In ACM Multimedia (Bristol, UK, Sept. 1998), pp. 303–308.

[25] Sen, S., Rexford, J., and Towsley, D. Proxy Prefix Caching for Multimedia Streams. In IEEE INFOCOM (New York, NY, Mar. 1999), pp. 1310–1319.

[26] Shenoy, P. J., and Vin, H. M. Efficient Striping Techniques for Multimedia File Servers. In Intl. Work. Network and Operating Systems Support for Audio and Video (St. Louis, MO, May 1997), pp. 25–36. An expanded version appears in Performance Evaluation Journal, vol. 38, June 1999, pg. 175–199.

[27] Wang, Y., and Du, D. H. Weighted Striping in Multimedia Servers. In *IEEE Multimedia Computing and Systems* (Ottawa, Canada, June 1997), pp. 102–109.

[28] Worthington, B. L., Ganger, G. R., Patt, Y. N., and Wilkes, J. On-Line Extraction of SCSI Disk Drive Parameters. In *ACM SIGMETRICS* (Ottawa, Canada, May 1995), pp. 146–156.

[29] Zhao, W., and Tripathi, S. K. Bandwidth-Efficient Continuous Media Streaming Through Optimal Multiplexing. In *ACM SIGMETRICS* (Atlanta, GA, June 1999), pp. 13–22.

[30] Zimmermann, R., and Ghandeharizadeh, S. Continuous Display Using Heterogeneous Disk-Subsystems. In *ACM Multimedia* (Seattle, WA, Nov. 1997), pp. 227–328.
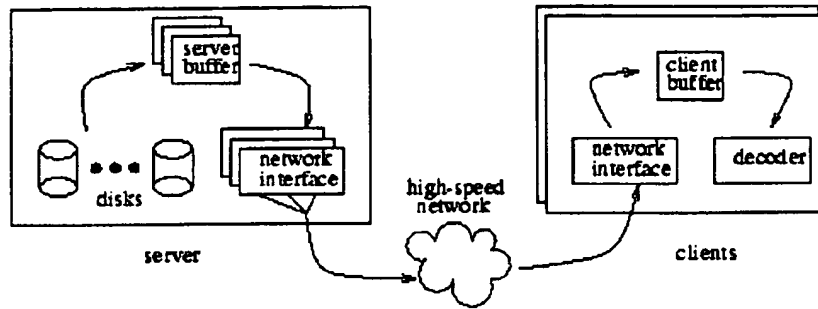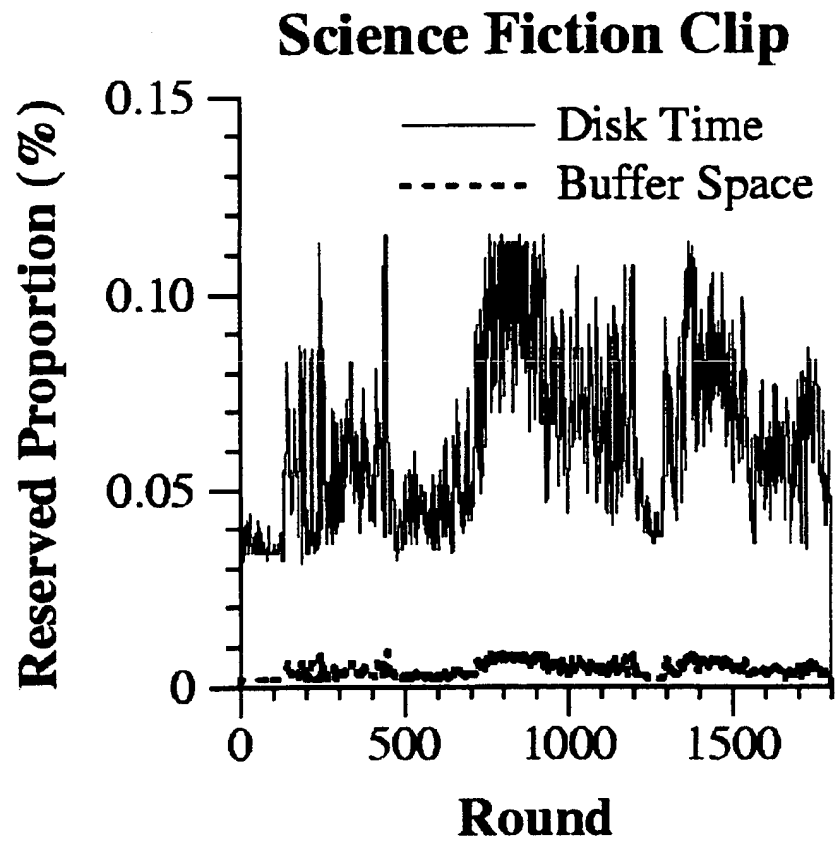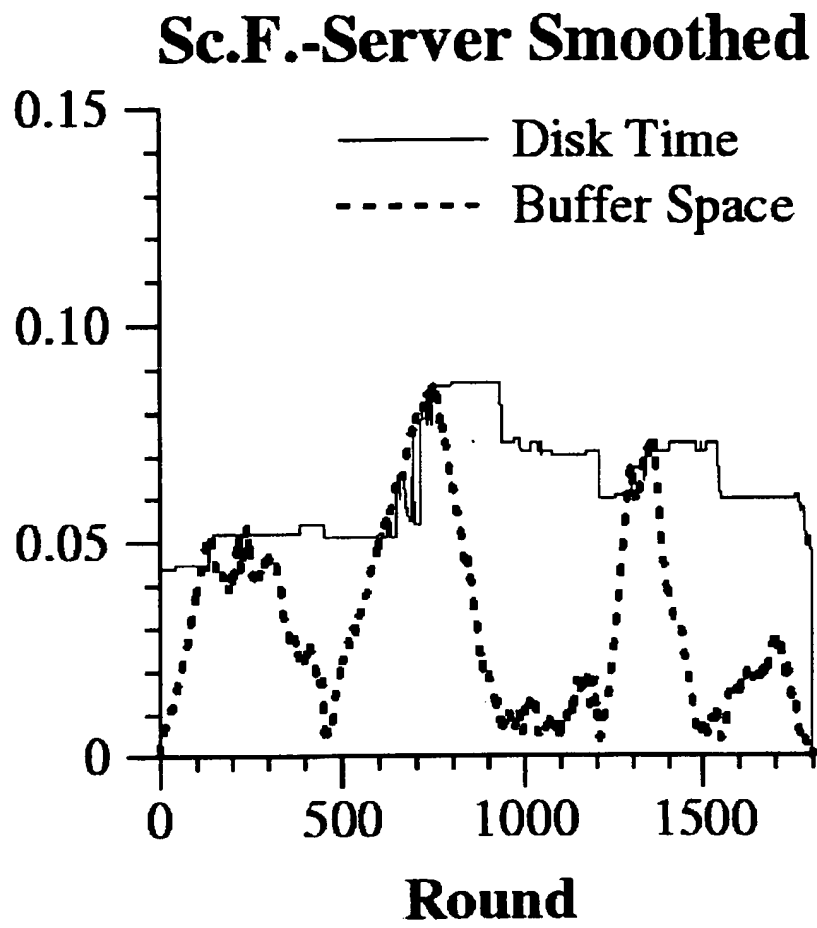
5

10

Figure 1

Figure 3(a)

Sc.F.-Server Smoothed

——— Disk Time

- - - - - Buffer Space

Figure 3(b)

Homogeneous Disks

Figure 4

Homogeneous Disks

Figure 5

Disk Time Requirements

Figure 6

Server Buffer Requirements

Figure 7

Science Fiction Clip

Figure 8(a)

Sc.F.-Server Smoothed
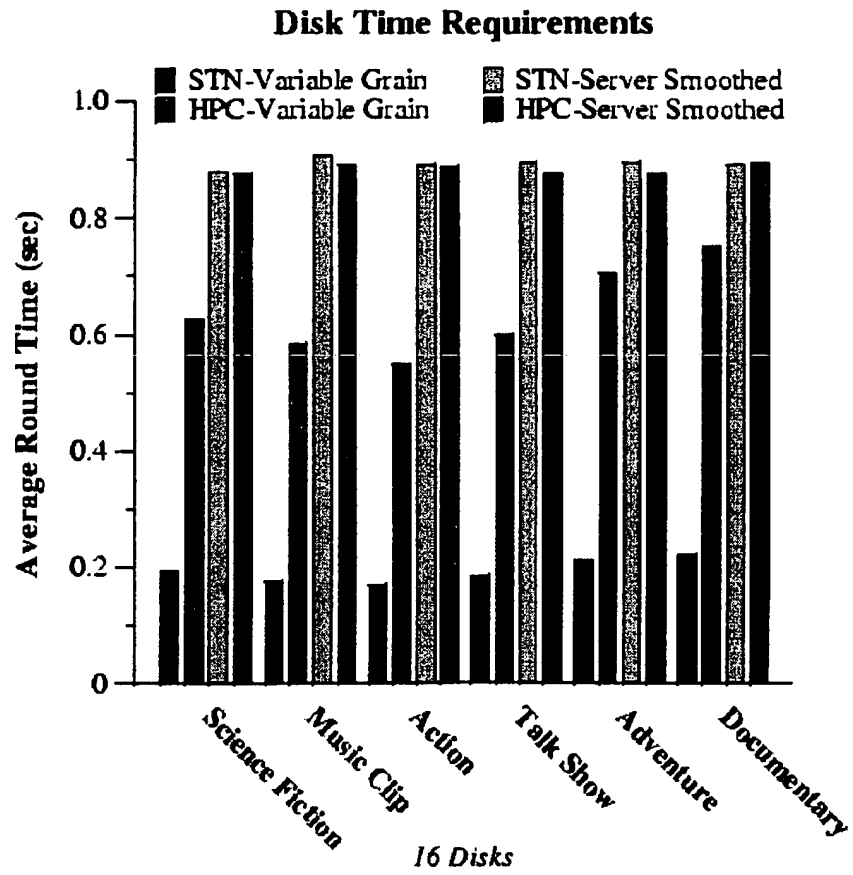
Figure 8(b)

Figure 9

Heterogeneous Disks

Figure 10

Disk Time Requirements
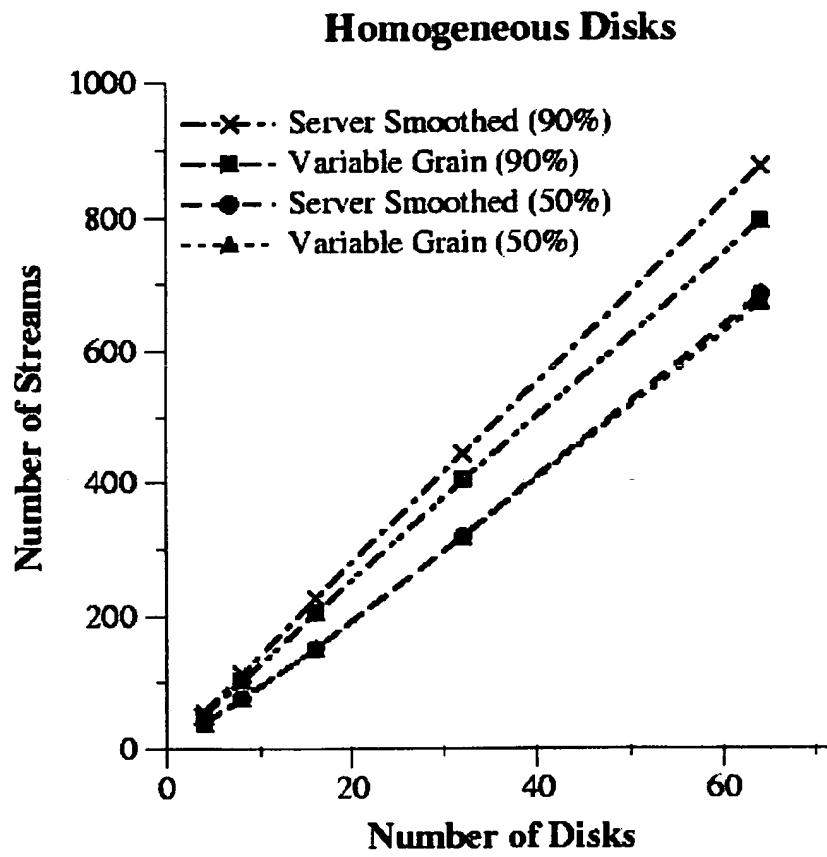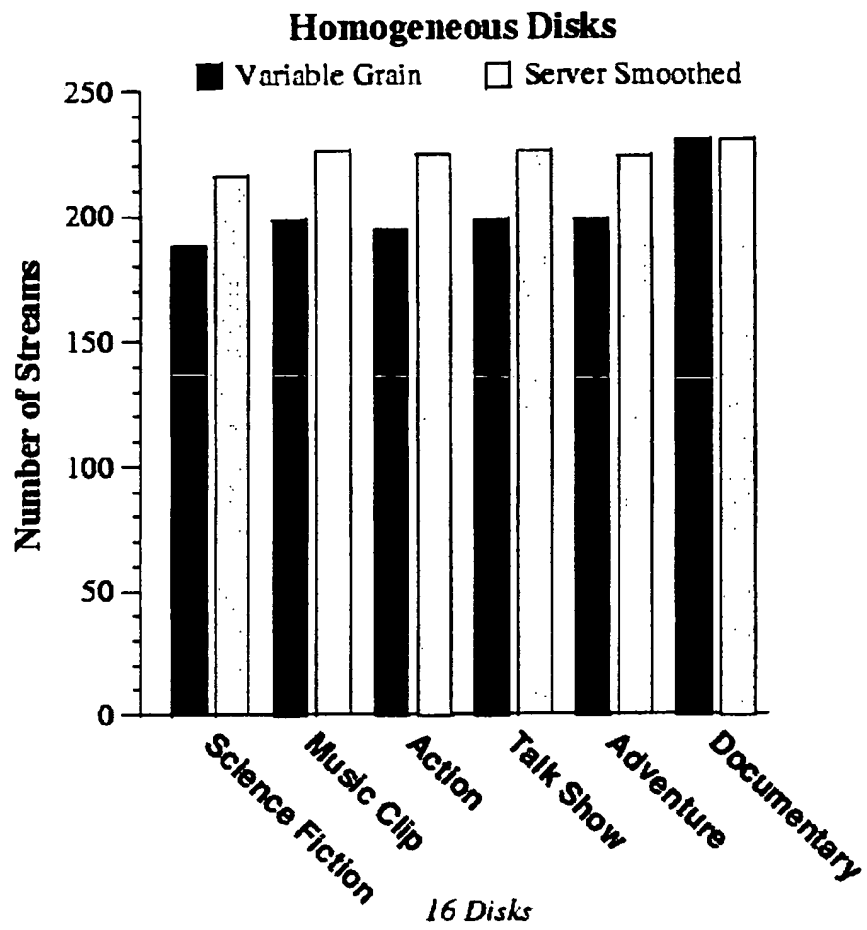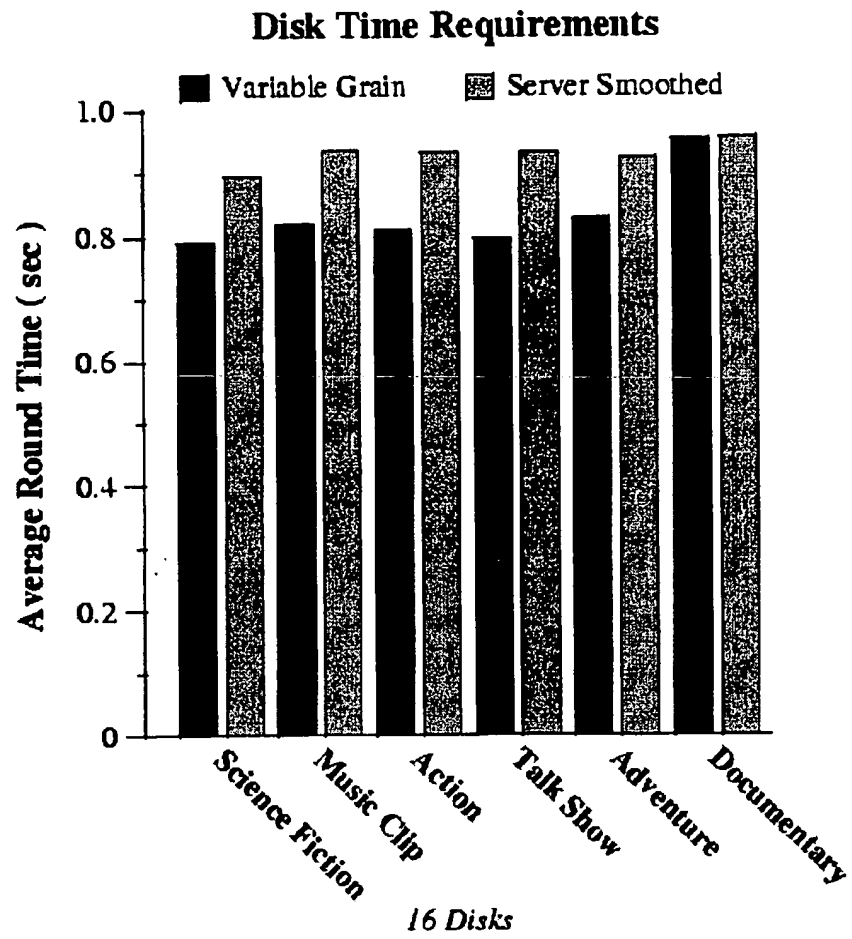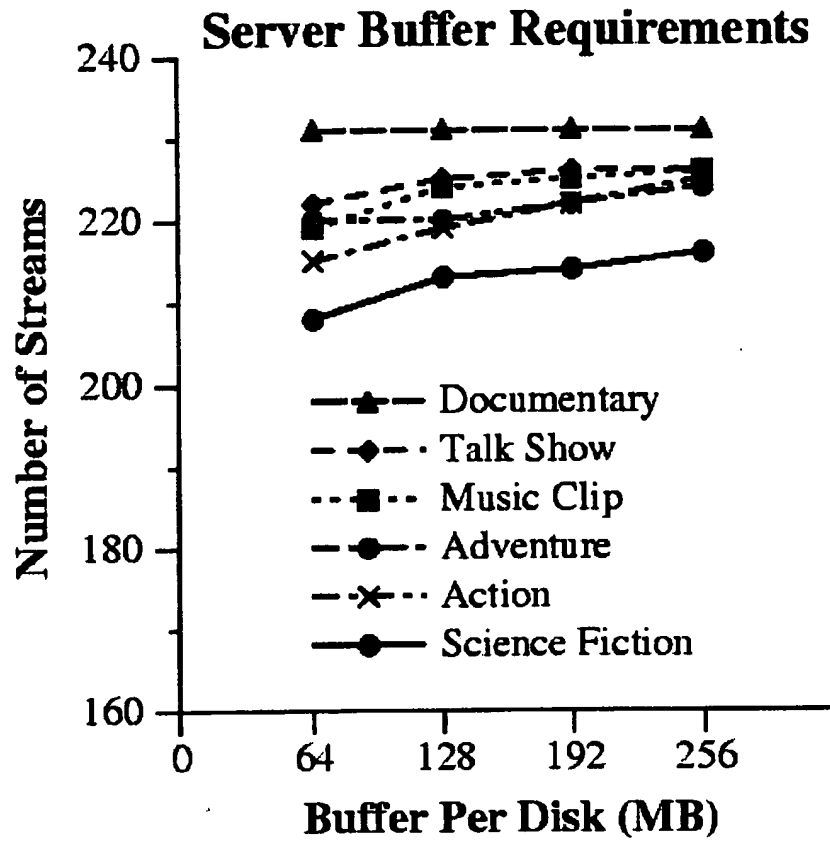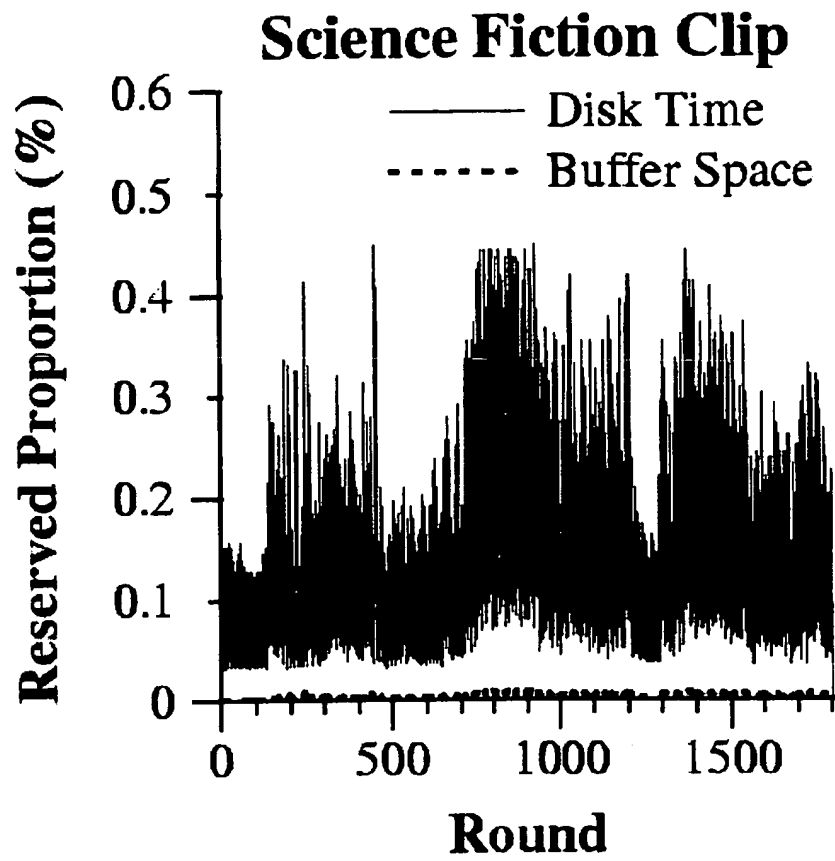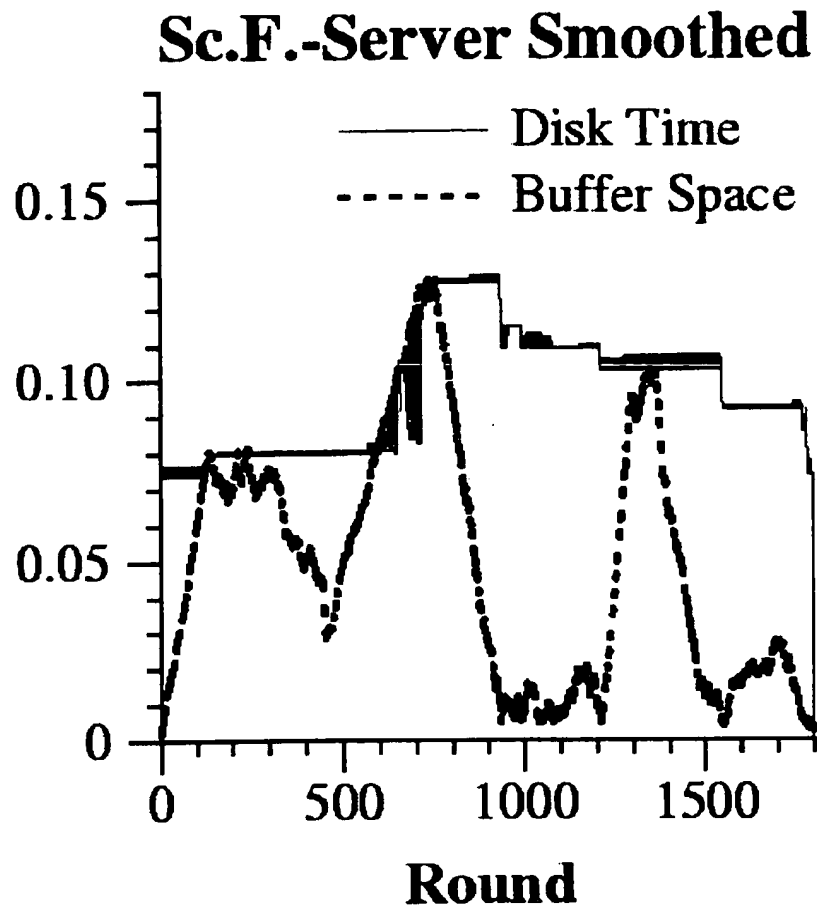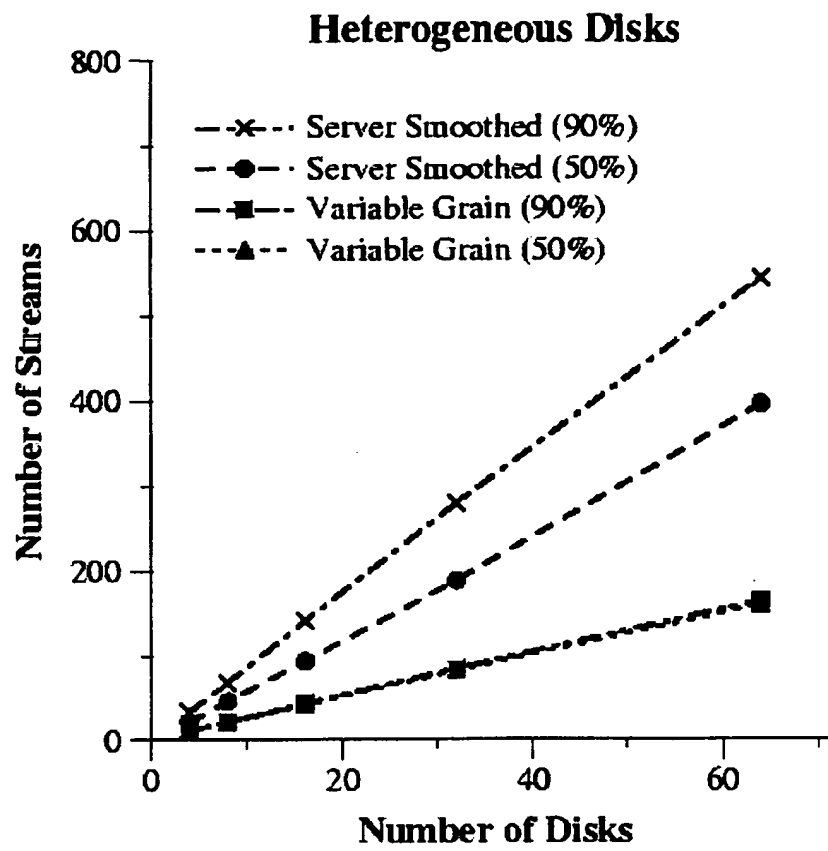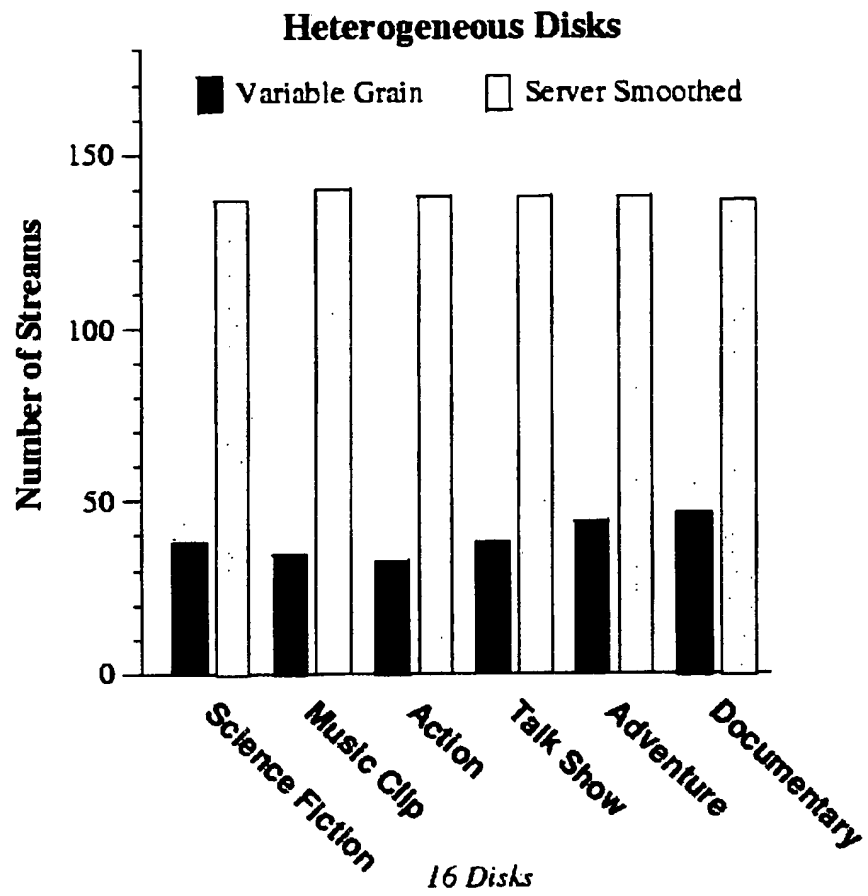
Figure 11

Figure 12

Figure 13